BIT Cheatsheet

Use a BIT when:

- You have to both calculate prefix sums and update values
- Usually goes with a sweep

A BIT has two possible uses. It can either:

1. Update (int x, int v): array[x] += v
2. Query (int x): return arr[1] + … + arr[x]

-or-

1. Update (int x, int v): arr[x] += v, arr[x+1] += v, …, arr[N] += v
2. Query (int x): return arr[x]

- Gold problems usually require the first option
- Note: Use 1-based indexing

Warning: Don't mix and match the two options

When thinking about BIT to use it in a problem:

- Think of a BIT as a magical array with fast prefix sum and update queries
- Don't consider the code for the update and query functions at all. Think about the problem conceptually and how the two operations you have available will help you.

Implementation:

C++:

```
int ft[N+1];
void update(int x, int v) {while(x<=N) ft[x]+=v, x+=(x&-x);}
int query (int x) { return x>0 ? ft[x]+query(x-(x&-x)):0;}
```

Java:

```
int[] ft = new int[N+1];
public static void update(int x, int v) {while(x<=N) {ft[x]+=v; x+=(x&-x);} }
public static int query (int x) { return x>0 ? ft[x]+query(x-(x&-x)):0;}
```

Runtime:

Update function: O(log N)
Query function: O(log N)